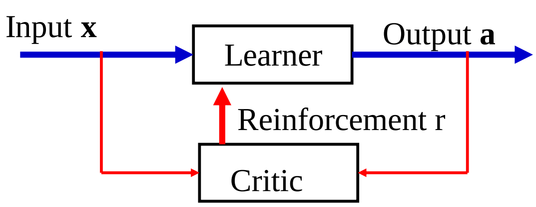
CS 2750 Machine Learning
Lecture 21b

Reinforcement learning

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Reinforcement learning

- **We want to learn a control policy:** $\pi : X \rightarrow A$
- We see examples of **x** (but outputs *a* are not given)
- Instead of *a* we get a feedback *r* (reinforcement, reward) from a **critic** quantifying how good the selected output was



- The reinforcements may not be deterministic
- **Goal:** find $\pi : X \rightarrow A$ with the best expected reinforcements

Gambling example

- **Game:** 3 biased coins
  - The coin to be tossed is selected randomly from the three coin options. The agent always sees which coin is going to be played next. The agent makes a bet on either a head or a tail with a wage of $1. If after the coin toss, the outcome agrees with the bet, the agent wins $1, otherwise it looses $1
- **RL model:**
  - **Input:** X – a coin chosen for the next toss,
  - **Action:** A – choice of head or tail the agent bets on,
  - **Reinforcements:** {1, -1}
- **A policy** $\pi : X \rightarrow A$

Example: $\pi :$ | **Coin1** ⟶ ***head***, **Coin2** ⟶ ***tail***, **Coin3** ⟶ ***head*** |

$\pi :$

⟶ head
⟶ tail
⟶ head

Gambling example

- **RL model:**
  - **Input:** X – a coin chosen for the next toss,
  - **Action:** A – choice of head or tail the agent bets on,
  - **Reinforcements:** {1, -1}
  - **A policy** $\pi :$ | **Coin1** ⟶ ***head***, **Coin2** ⟶ ***tail***, **Coin3** ⟶ ***head*** |
- **Learning goal: find the optimal policy**

$$\pi^* : X \rightarrow A$$

maximizing future expected profits

$$E(\sum_{t=0}^{T} \gamma^t r_t) \qquad 0 \le \gamma < 1$$

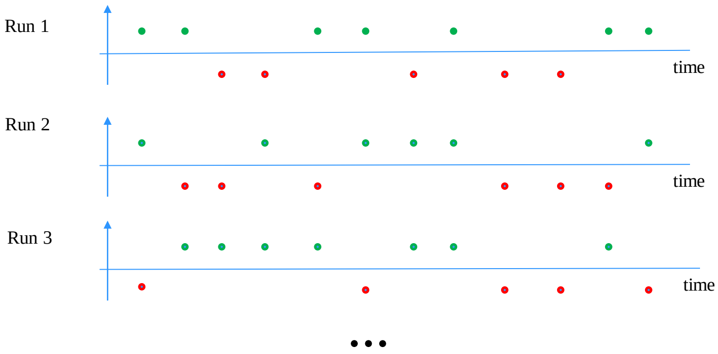
a discount factor = present value of money

$\pi^* :$

⟶ ?
⟶ ?
⟶ ?

Expected rewards

- **Expected rewards for** $\pi : X \to A$



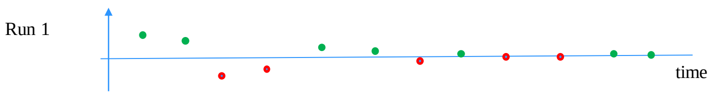
$E(\sum_{t=0}^{T} r_t)$ Expectation over many possible reward trajectories for $\pi : X \to A$

Expected discounted rewards

- **Expected discounting rewards for** $\pi : X \to A$
- **Discounting with** $0 \leq \gamma < 1$ (future value of money)

No discounting:

Discounting



$E(\sum_{t=0}^{T} \gamma^t r_t)$ Expectation over many possible discounted reward trajectories for $\pi : X \to A$

RL learning: objective functions

- **Objective:**
  Find a mapping $\pi^* : X \rightarrow A$
  That maximizes some combination of future reinforcements (rewards) received over time
- **Valuation models (quantify how good the mapping is):**
  - **Finite horizon models**

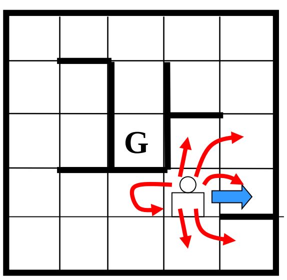
    $E(\sum_{t=0}^{T} r_t)$ Time horizon: $T > 0$

    $E(\sum_{t=0}^{T} \gamma^t r_t)$ Discount factor: $0 \le \gamma < 1$
  - **Infinite horizon discounted model**

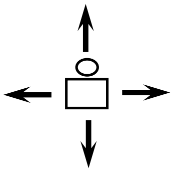
    $E(\sum_{t=0}^{\infty} \gamma^t r_t)$ Discount factor: $0 \le \gamma < 1$
  - **Average reward** $\lim_{T \rightarrow \infty} \frac{1}{T} E(\sum_{t=0}^{T} r_t)$

Agent navigation example

- **Agent navigation in the maze**:
  - 4 moves in compass directions
  - Effects of moves are stochastic – we may wind up in other than intended location with a non-zero probability
  - **Objective:** learn how to reach the goal state in the shortest expected time

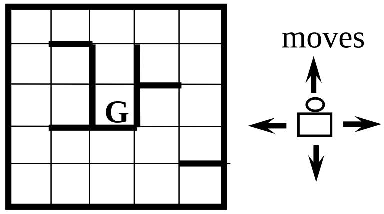


moves



Agent navigation example

- **The RL model:**
  - **Input:** X – a position of an agent
  - **Output:** A –the next move
  - **Reinforcements:** R
    - -1 for each move
    - +100 for reaching the goal
  - **A policy:** $\pi : X \rightarrow A$



$\pi :$

Position 1 →	***right***
Position 2 →	***right***
…	
Position 20 →	***left***

- **Goal:** **find the policy maximizing future expected rewards**

$$E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \qquad 0 \le \gamma < 1$$

Exploration vs. Exploitation in RL

- The (learner) actively interacts with the environment:
  - At the beginning the learner does not know anything about the environment
  - It gradually gains the experience and learns how to react to the environment
- **Dilemma (exploration-exploitation):**
  - After some number of steps, should I select the best current choice (**exploitation**) or try to learn more about the environment (**exploration**)?
  - **Exploitation** may involve the selection of a sub-optimal action and prevent the learning of the optimal choice
  - **Exploration** may spend to much time on trying bad currently suboptimal actions

Effects of actions on the environment

Effect of actions on the environment (next input **x** to be seen)

- **No effect.** The distribution over possible **x** is fixed and independent of past actions. The rewards received depend only on the state **x** and action chosen. The are seen after the action.
- **Actions may effect the environment** and next inputs **x**. The distribution of **x** can change due to past actions; the rewards related to the action can be seen with some delay.

Leads to two forms of **reinforcement learning:**

- **Learning with immediate rewards**
  - **3 coin example** 

- **Learning with delayed rewards**
  - **Agent navigation example**; 

    move choices affect the state of the environment (position changes), a big reward at the goal state is delayed

RL with immediate rewards

- **Game:** 3 biased coins
  - The coin to be tossed is selected randomly from the three coin options. The agent always sees which coin is going to be played next. The agent makes a bet on either a head or a tail with a wage of \$1. If after the coin toss, the outcome agrees with the bet, the agent wins \$1, otherwise it looses \$1
- **RL model:**
  - **Input:** X – a coin chosen for the next toss
  - **Action:** A – head or tail the agent bets on
  - **Reinforcements:** {1, -1} (\$1 either won or lost)
- **Learning goal: find the optimal policy** $\pi^* : X \rightarrow A$

  maximizing the future expected profits over time

  $E(\sum_{t=0}^{\infty} \gamma^t r_t)$ $\quad 0 \leq \gamma < 1$ a discount factor

RL with immediate rewards

- **Expected reward** $E(\sum_{t=0}^{\infty} \gamma^t r_t) \qquad 0 \le \gamma < 1$
- **Immediate reward case:**
  - Reward depends only on **x** and the action choice
  - The action does not affect the environment and hence future inputs (states) and future rewards:

$$E(\sum_{t=0}^{\infty} \gamma^t r_t) = E(r_0) + E(\gamma r_1) + E(\gamma^2 r_2) + \dots$$

$r_0, r_1, r_2 \dots$ Rewards for every step of the game

  - Expected one step reward for input **x (coin to play next)** and the choice *a* : $R(\mathbf{x}, a)$

RL with immediate rewards

Immediate reward case:

- Reward for input **x** and the action choice *a* may vary
- **Expected reward for the input x and choice *a*:** $R(\mathbf{x}, a)$
  - For the coin bet problem it is:

$$R(\mathbf{x}, a_i) = \sum_j r(\omega_j \mid a_i, \mathbf{x}) P(\omega_j \mid \mathbf{x}, a_i)$$

ω_j : an outcome of the coin toss x

$r(\omega_j \mid a_i, \mathbf{x})$: reward for an outcome and the bet made on **x**

- **Expected one step reward for a strategy**

$$R(\pi) = \sum_x R(\mathbf{x}, \pi(\mathbf{x})) P(\mathbf{x}) \qquad \pi : X \to A$$

$R(\pi)$ is the expected reward for $r_0, r_1, r_2 \dots$

RL with immediate rewards

- **Expected reward**

$$E(\sum_{t=0}^{\infty} \gamma^t r_t) = E(r_0) + E(\gamma r_1) + E(\gamma^2 r_2) + \ldots$$

- **Optimizing the expected reward :**

$$\max_{\pi} E(\sum_{t=0}^{\infty} \gamma^t r_t) = \max_{\pi} \sum_{t=0}^{\infty} \gamma^t E(r_t) = \max_{\pi} \sum_{t=0}^{\infty} \gamma^t R(\pi) = \max_{\pi} R(\pi)(\sum_{t=0}^{\infty} \gamma^t)$$

$$= (\sum_{t=0}^{\infty} \gamma^t) \max_{\pi} R(\pi)$$

$$\max_{\pi} R(\pi) = \max_{\pi} \sum_{x} R(\mathbf{x}, \pi(\mathbf{x})) P(\mathbf{x}) = \sum_{x} P(\mathbf{x}) [\max_{\pi(\mathbf{x})} R(\mathbf{x}, \pi(\mathbf{x}))]$$

- **Optimal strategy:** $\pi^* : X \rightarrow A$

$$\pi^*(\mathbf{x}) = \arg\max_{a} R(\mathbf{x}, a)$$

RL with immediate rewards

- **We know that** $\pi^*(\mathbf{x}) = \arg\max_{a} R(\mathbf{x}, a)$
- **Problem:** In the RL framework we do not know $R(\mathbf{x}, a)$
  - The expected reward for performing action *a* at input **x**
- **How to estimate** $R(\mathbf{x}, a)$ **?**

RL with immediate rewards

- **Problem:** In the RL framework we do not know $R(\mathbf{x}, a)$
  - The expected reward for performing action *a* at input **x**
- **Solution:**
  - For each input **x** try different actions *a*
  - Estimate $R(\mathbf{x}, a)$ using the average of observed rewards

$$\tilde{R}(\mathbf{x}, a) = \frac{1}{N_{x,a}} \sum_{i=1}^{N_{x,a}} r_i^{x,a}$$

  - Action choice $\pi(\mathbf{x}) = \arg\max_a \tilde{R}(\mathbf{x}, a)$
  - Accuracy of the estimate: statistics (Hoeffding's bound)

$$P\left(\left|\tilde{R}(\mathbf{x}, a) - R(\mathbf{x}, a)\right| \geq \varepsilon\right) \leq \exp\left[-\frac{2\varepsilon^2 N_{x,a}}{(r_{\max} - r_{\min})^2}\right] \leq \delta$$

  - Number of samples: $N_{x,a} \geq \frac{(r_{\max} - r_{\min})^2}{2\varepsilon^2} \ln \frac{1}{\delta}$

RL with immediate rewards

- **On-line (stochastic approximation)**
  - An alternative way to estimate $R(\mathbf{x}, a)$
- **Idea:**
  - choose action *a* for input **x** and observe a reward $r^{x,a}$
  - Update an estimate in every step *i*

$$\tilde{R}(\mathbf{x}, a)^{(i)} \leftarrow (1 - \alpha(i))\tilde{R}(\mathbf{x}, a)^{(i-1)} + \alpha(i)\, r_i^{x,a}$$

$\alpha(i)$ - a learning rate

- **Convergence property:** The approximation converges in the limit for an appropriate learning rate schedule.
- Assume: $\alpha(n(x, a))$ - is a learning rate for *n*th trial of (*x*,*a*) pair
- Then the converge is assured if:

1. $\sum_{i=1}^{\infty} \alpha(i) = \infty$
2. $\sum_{i=1}^{\infty} \alpha(i)^2 < \infty$

RL with immediate rewards

- At any step in time *i* during the experiment we have estimates of expected rewards for each (*coin, action*) pair:

$$\tilde{R}(coin1, head)^{(i)}$$
$$\tilde{R}(coin1, tail)^{(i)}$$
$$\tilde{R}(coin2, head)^{(i)}$$
$$\tilde{R}(coin2, tail)^{(i)}$$
$$\tilde{R}(coin3, head)^{(i)}$$
$$\tilde{R}(coin3, tail)^{(i)}$$

- Assume the next coin to play in step $(i+1)$ is coin 2 and we pick head as our bet. Then we update $\tilde{R}(coin2, head)^{(i+1)}$ using the observed reward and one of the update strategy above, and keep the reward estimates for the remaining (coin, action) pairs unchanged, e.g. $\tilde{R}(coin2, tail)^{(i+1)} = \tilde{R}(coin2, tail)^{(i)}$

Exploration vs. Exploitation

- In the RL framework
  - the (learner) actively interacts with the environment and **choses the action** to play for the current input **x**
  - Also at any point in time it has an estimate of $\tilde{R}(\mathbf{x}, a)$ for any (*input,action*) pair
- **Dilemma for choosing the action to play for x:**
  - Should the learner choose the current best choice of action (exploitation) $$\hat{\pi}(\mathbf{x}) = \arg\max_{a \in A} \tilde{R}(\mathbf{x}, a)$$
  - Or choose some other action *a* which may help to improve its $\tilde{R}(\mathbf{x}, a)$ estimate (exploration)

  This dilemma is called **exploration/exploitation dilemma**
- **Different exploration/exploitation strategies exist**

Exploration vs. Exploitation

- **Uniform exploration:** Exploration parameter $0 \le \varepsilon \le 1$
  - Choose the "current" best choice with probability $1-\varepsilon$

$$\hat{\pi}(\mathbf{x}) = \arg\max_{a \in A} \tilde{R}(\mathbf{x}, a)$$

  - All other choices are selected with a uniform probability $\dfrac{\varepsilon}{|A|-1}$
- **Boltzman exploration**
  - The action is chosen randomly but proportionally to its current expected reward estimate

$$p(a \mid \mathbf{x}) = \frac{\exp\left[\tilde{R}(x,a)/T\right]}{\sum_{a' \in A} \exp\left[\tilde{R}(x,a')/T\right]}$$

T – is temperature parameter. **What does it do?**